
VL53L4CD ultralow power with accuracy

Introduction

The ultralow power (ULP) driver for the VL53L4CD is designed to check proximity without using much power. On occasion, a user might want to use low power, but ensure that when a target is detected, get a more accurate distance reading.

This document describes two functions added to the ULP driver to accomplish this as well as some proposed pseudocode on how to use the functions.

This can be used to replace the functionality of the VL6180. The VL6180 was unique in its ability to end the range quickly if few photons were returned. This feature saved a lot of power. This software emulates that ability.

1 Theory of operation

This type of low-power operation helps save battery life when doing short-distance detects. The 5 ms ranges are done perhaps 2-4 times per second. Only when a target is detected within some specified range is it worth spending the energy to get an accurate measure.

2 Prototypes

```

/**
 * @brief This function configures the sensor for a longer collect.
 * It enables a more complete - and accurate - range.
 * @param (uint16_t) dev: Device address.
 * @param (uint16_t) verification_timing : amount of time to integrate.
 * [0,255] although 95 is a good number about 17ms.
 * @return (uint8_t) status: 0 if OK.
 */

uint8_t VL53L4CD_ULP_Target_Verification_Mode(
    uint16_t dev,
    uint16_t verification_timing);

/**
 * @brief This function returns the sensor to the user's very short
 * integration mode saving power
 * It limits the PHASE DETECT, VHV and DSS
 * Call this function after calling
 * VL53L4CD_ULP_Target_Verification_Mode(), and ranging once
 * to return to the lowest power consuming mode.
 * @param (uint16_t) dev : Device address.
 * @param (uint16_t) detection_timing : low power integration timing.
 * [0,255] but 1 is a good number, about 5ms.
 * @param (uint8_t) enable_interrupt_only_below_threshold : If this
 * flag is set to 0, the sensor is programmed with the first
 * configuration (raise an interrupt when a new measurement is
 * ready). Else, the sensor is programmed to raise an interrupt ONLY
 * when a valid distance below the threshold is reported.
 * @return (uint8_t) status : 0 if OK.
 */

uint8_t VL53L4CD_ULP_Target_Detection_Mode(
    uint16_t dev,
    uint16_t detection_timing,
    uint8_t enable_interrupt_only_below_threshold);

```

3 Pseudocode

With the sensor running with the ULP driver and two timings defined:

- DETECTION_TIME – A small positive integer. 1 equates to about 5ms.
- VERIFICATON_TIME – A larger positive integer. 95 equates to about 17ms.

Run the following code:

```
{
  /* wait for an interrupt - something is close*/
  __WFI ();
  /* Set up for precision ranging */
  status = VL53L4CD_ULP_StopRanging(dev);

  /* Configure for long collect and start */
  status |= VL53L4CD_ULP_Target_Verification_Mode(dev, VERIFICAION_TIME);
  status |= VL53L4CD_ULP_StartRanging(dev);

  /* wait for high precision range to complete */
  __WFI ();

  /* stop the sensor and get the data */
  status = VL53L4CD_ULP_StopRanging(dev);
  status |= VL53L4CD_ULP_DumpDebugData(
    dev,
    &measurement_status,
    &estimated_distance_mm,
    &sigma_mm,
    &signal_kcps,
    &ambient_kcps);

  /* Return to Low Power Detection mode */
  status = VL53L4CD_ULP_Target_Detection_Mode(dev, DETECTION_TIME, 1);

  /* Wait to get to the next inter-measurement period */
  HAL_Delay(460);

  /* Restart the sensor */
  status |= VL53L4CD_ULP_StartRanging(dev);
} while (1);
```

4 Power consumption plot

Figure 1. Power consumption plot



In the plot above the first 5 ms range is the proximity detection of the ULP driver. The short, 1 uA section between the ranges is the I2C access required to reconfigure the sensor. The large 17 ms range on the right is the high-precision range.

The power figures are approximate and depend on the amount of ambient light and the number of returned photons.

5 Example code

This code is divided into two sections for clarity.

5.1 Initialization

```
#define DETECTION_TIME 1
#define VERIFICATION_TIME 95
#define INT_MODE 1
#define INTERMEASUREMENT_TIME 500

uint8_t measurement_status;
uint16_t estimated_distance_mm;
uint16_t sigma_mm;
uint16_t signal_kcps;
uint16_t ambient_kcps;

/* Program sensor to raise an interrupt ONLY below 300mm */
status = VL53L4CD_ULP_SetInterruptConfiguration(dev, 300, INT_MODE);

/* Program a ranging frequency */
status = VL53L4CD_ULP_SetInterMeasurementInMs(dev, INTERMEASUREMENT_TIME);

/* Enable 16 SPADS. Choosing more requires more power, but can range farther */
status = VL53L4CD_ULP_SetROI(dev, 16);

/*****
 *
 * Starting measurements
 *****/
printf("Ranging starts. Put a hand close to the sensor to raise an interrupt... \n");
status = VL53L4CD_ULP_StartRanging(dev);
```

5.2 Operation

```

do
{
    /* wait for an interrupt */
    IntCount=0;
    while(IntCount ==0 ){
        __WFI();    // Wait for the short-range interrupt
    }
    IntCount=0;

    /* Got a quick detect */
    status = VL53L4CD_ULP_StopRanging(dev);
    VL53L4CD_ULP_ClearInterrupt(dev); /* Clear HW interrupt to restart measurements */
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_0, GPIO_PIN_SET);

    /* Set the Sensor to enter higher power Target Verification mode and range once */
    status |= VL53L4CD_ULP_Target_Verification_Mode(dev, VERIFICATION_TIME);

    status |= VL53L4CD_ULP_StartRanging(dev);
    VL53L4CD_ULP_ClearInterrupt(dev);

    IntCount=0;
    while(IntCount ==0 ){
        __WFI();    // Wait for Long-range interrupt
    }
    IntCount=0;

    status = VL53L4CD_ULP_StopRanging(dev);
    VL53L4CD_ULP_ClearInterrupt(dev);

    status |= VL53L4CD_ULP_DumpDebugData(
        dev,
        &measurement_status,
        &estimated_distance_mm,
        &sigma_mm,
        &signal_kcps,
        &ambient_kcps);
    if (status) printf("%d : * ERROR *\n", __LINE__);

    /* Go back to low-power target detection mode */
    status = VL53L4CD_ULP_Target_Detection_Mode(dev, DETECTION_TIME, INT_MODE);

    printf("\n status = %2d, dist = %3d, Signal - %5d, Sigma = %2d, Amb = %5d\n",
        measurement_status,
        estimated_distance_mm,
        signal_kcps,
        sigma_mm,
        ambient_kcps);

    HAL_Delay(470); // 460 is the Inter-measurement period minus the ranging and print time.

    status |= VL53L4CD_ULP_StartRanging(dev);
    if (status) printf("%d : * ERROR *\n", __LINE__);
    VL53L4CD_ULP_ClearInterrupt(dev);
    IntCount=0;
}while(1);

```

6 Usage

Refer to the example in the following package:

- [STSW-IMG049_ECE](#): Example code for Time-of-Flight migration from VL6180 to VL53L4CD, a high-accuracy, and low-power proximity sensor.

Revision history

Table 1. Document revision history

Date	Version	Changes
31-Jul-2024	1	Initial release

Contents

1	Theory of operation	2
2	Prototypes	3
3	Pseudocode	4
4	Power consumption plot	5
5	Example code	6
5.1	Initialization	6
5.2	Operation	7
6	Usage	8
	Revision history	9

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2024 STMicroelectronics – All rights reserved